WHAT IS CLAIMED IS:

1.   A floating point total order comparator circuit for comparing a first floating

point operand and a second floating point operand, comprising:

a first analysis circuit for determining a format of the first floating point operand

based upon floating point status information encoded within the first floating point operand;

a second analysis circuit for determining a format of the second floating point operand

based upon floating point status information encoded within the second floating point

operand; and

a result generator circuit coupled to the analysis circuits for producing a result

indicating a total order comparative relationship between the first floating point operand and

the second floating point operand based on the format of the first floating point operand and

the format of the second floating point operand.

2.   The floating point total order comparator circuit of claim 1, further

comprising:

a first operand buffer coupled to the first analysis circuit, for supplying the first

floating point operand to the first analysis circuit; and

a second operand buffer coupled to the second analysis circuit, for supplying the

second floating point operand to the second analysis circuit.

3.   The floating point total order comparator circuit of claim 1, wherein the result

indicating the total order comparative relationship between the first floating point operand

and the second floating point operand comprises at least one of the group comprising: the

170646-1

first operand is less than the second operand, the first operand is greater than the second

operand, and the first operand is equal to the second operand.

4.   The floating point total order comparator circuit of claim 1, wherein the

format is from a group comprising: not-a-number (NaN), infinity, normalized, denormalized,

invalid operation, overflow, underflow, division by zero, positive zero, negative zero, exact,

and inexact.

5.   The floating point total order comparator circuit of claim 4 wherein the format

represents one of a positive overflow (+OV) and a negative overflow (-OV).

6.   The floating point total order comparator circuit of claim 4 wherein the format

represents one of a positive underflow (+UN) and a negative underflow (-UN).

7.   The floating point total order comparator circuit of claim 4, wherein the

format represents one of a positive infinity and a negative infinity.

8.   The floating point total order comparator circuit of claim 1, wherein the result

is used to condition an outcome of a floating point instruction.

9.   The floating point total order comparator circuit of claim 1, wherein the

format represents a combination of at least two of a group comprising: not-a-number (NaN),

infinity, normalized, denormalized, invalid operation, overflow, underflow, division by zero, positive zero, negative zero exact, and inexact.

10.   The floating point total order comparator circuit of claim 4, wherein the total order comparative relationship represents one of a group comprising:

a relationship between the first floating point operand having the NaN format and the second floating point operand having the NaN format;

a relationship between having first floating point operand having the NaN format and the second floating point operand not having the NaN format; and

a relationship between having a first floating point operand not having the NaN format and the second floating point operand having the NaN format.

11.   The floating point total order comparator circuit of claim 4, wherein the total order comparative relationship indicates that:

the first floating point operand having the positive zero format is greater than the second floating point operand having the negative zero format.

12.   The floating point total order comparator circuit of claim 1, wherein the total order comparative relationship indicates that if the first floating point operand has the positive NaN format and the second floating point operand has the negative NaN format, and the first floating point operand contains the same value as the second floating point operand, ignoring the encoded status information in each, then the first floating point operand is greater than the second floating point operand.

170646-1

13.   The floating point total order comparator circuit of claim 1, wherein the total order comparative relationship indicates that if the first floating point operand has the NaN format and the second floating point operand has the NaN format, then the one of the two floating point operands containing a larger value in a fraction field of the two floating point operands, ignoring the encoded status information in each, is greater than the other of the two floating point operands, regardless of a respective sign bit of each floating point operand.

14.   The floating point total order comparator circuit of claim 1, wherein the total order comparative relationship indicates that if the first floating point operand has the NaN format and the second floating point operand has the NaN format, and the two floating point operands contain the same value in a fraction field, an exponent field, and a sign bit of the respective floating point operands, then the one of the two floating point operands containing a larger value in the encoded status information in each is greater than the other of the two floating point operands.

15.   A method for comparing a first floating point operand and a second floating point operand according to a predefined total order comparative relationship, comprising:

receiving the first floating point operand and the second floating point operand;

determining a first floating point format of the first floating point operand from floating point status information encoded within the first floating point operand;

determining a second floating point format of the second floating point operand from floating point status information encoded within the second floating point operand; and

generating a result indicating the total order comparative relationship between the first

floating point operand and the second floating point operand based on the first floating point

format and the second floating point format.

16.   The method of claim 15, further comprising:

conditioning the outcome of a floating point instruction based upon the result

generated.

17.   The method of claim 15, wherein the first floating point format and the second

floating point format are from a group comprising: not-a-number (NaN), infinity,

normalized, denormalized, invalid operation, overflow, underflow, division by zero, positive

zero, negative zero, exact, and inexact.

18.   The method of claim 17, wherein the group further comprises: positive

overflow (+OV) and negative overflow (-OV).

19.   The method of claim 17, wherein the group further comprises: positive

overflow (+UN) and negative overflow (-UN).

20.   The method of claim 17, wherein the group further comprises: positive

infinity and negative infinity.

170646-1

-47-

21.   The method of claim 17, wherein the total order comparative relationship indicates represents one of a group comprising:

a relationship between the first floating point operand having the NaN format and the second floating point operand having the NaN format;

a relationship between the first floating point operand having the NaN format and the second floating point operand not having the NaN format; and

a relationship between the first floating point operand not having the NaN format and the second floating point operand having the NaN format.

22.   The method of claim 17, wherein the total order comparative relationship indicates that one of the two floating point operands having the positive zero format is greater than another of the two floating point operands having the negative zero format.

23.   The method of claim 17, wherein the total order comparative relationship indicates that if the first floating point operand has the positive NaN format and the second floating point operand has the negative NaN format, and the first floating point operand contains the same value as the second floating point operand, ignoring the encoded status information in each, then the first floating point operand is greater than the second floating point operand.

24.   The method of claim 17, wherein the total order comparative relationship indicates that if the first floating point operand has the NaN format and the second floating point operand has the NaN format, then the one of the two floating point operands containing

a larger value in a fraction field of the two floating point operands, ignoring the encoded

status information in each, is greater than the other of the two floating point operands,

regardless of a respective sign bit of each floating point operand.

25.   The method of claim 17, wherein the total order comparative relationship

indicates that if the first floating point operand has the NaN format and the second floating

point operand has the NaN format, and the two floating point operands contain the same

value in a fraction field, an exponent field, and a sign bit of the respective floating point

operands, then the one of the two floating point operands containing a larger value in the

encoded status information in each is greater than the other of the two floating point

operands.

26.   A computer-readable medium on which is stored a set of instructions for

comparing a first floating point operand and a second floating point operand according to a

predefined total order comparative relationship, which when executed perform steps

comprising:

receiving the first floating point operand and the second floating point operand related

to a floating point instruction;

determining a first floating point format of the first floating point operand from

floating point status information encoded within the first floating point operand;

determining a second floating point format of the second floating point operand from

floating point status information encoded within the second floating point operand; and

generating a result indicating the total order comparative relationship between the first floating point operand and the second floating point operand based on the first floating point format and the second floating point format.

27. The computer-readable medium of claim 26, further comprising:

conditioning the outcome of a floating point instruction based upon the result generated.

28. The computer-readable medium of claim 26, wherein the first floating point format and the second floating point format are from a group comprising: not-a-number (NaN), infinity, normalized, denormalized, invalid operation, overflow, underflow, division by zero, positive zero, negative zero, exact, and inexact.

29. The computer-readable medium of claim 28, wherein the total order comparative relationship represents on of a group comprising:

a relationship between the first floating point operand having the NaN format and the second floating point operand having the NaN format;

a relationship between the first floating point operand having the NaN format and the second floating point operand not having the NaN format; and

a relationship between the first floating point operand not having the NaN format and the second floating point operand having the NaN format.

30.   The computer-readable medium of claim 28, wherein the total order

comparative relationship indicates that one of the two floating point operands having the

positive zero format is greater than the other of the two floating point operands having the

negative zero format.

31.   The computer-readable medium of claim 28, wherein the total order

comparative relationship indicates that if the first floating point operand has the positive NaN

format, and the second floating point operand has the negative NaN format, and the first

floating point operand contains the same value as the second floating point operand, ignoring

the encoded status information in each, then the first floating point operand is greater than the

second floating point operand.

32.   The computer-readable medium of claim 28, wherein the total order

comparative relationship indicates that if the first floating point operand has the NaN format,

and the second floating point operand has the NaN format, then the one of the two floating

point operands containing a larger value in a fraction field of the two floating point operands,

ignoring the encoded status information in each, is greater than the other of the two floating

point operands, regardless of a respective sign bit of each floating point operand.

33.   The computer-readable medium of claim 28, wherein the total order

comparative relationship indicates that if the first floating point operand has the NaN format,

and the second floating point operand has the NaN format, and the two floating point

operands contain the same values in a fraction field, an exponent field, and a sign bit of the

170646-1

-51-

respective floating point operands, then the one of the two floating point operands containing

a larger value in the encoded status information in each is greater than the other of the two

floating point operands.

170646-1